POLITECNICO
MILANO 1863

Project Plan

version 1.0

22nd January 2017

Authors:
Davide Moneta (808686)
Federico Oldani (806337)
Luca Oppedisano (878301)

Index

1 Introduction

1.1 Purpose

This document consists in the Project Planning Document. Here we will try to estimate the complexity of the system that will ultimately be implemented and the costs and efforts of the related development process.

Based on this figures it will be possible to settle a budget, to define a schedule and to allocate the resources in the best way.

We will quantify the size and complexity of the code using Function Points and COCOMO II, we will define a schedule and allocate the resources with GANTT diagrams and we will think of a strategy to mitigate any unwanted circumstance that may occur.

1.2 Scope

We will design a software to manage the functionalities of PowerEnJoy: a new car sharing service. PowerEnJoy offers exclusively electric cars that allow the users to move around respecting the environment.

Using this platform customers are able to find and reserve one of our cars within a certain distance from their current location or a specified address. When they reach the chosen car, they can unlock it and drive it.

Users are charged with a per-minute fee, but the company incentivizes virtuous behaviors by offering a discount at the end of a trip if certain conditions are met.

1.3 Definitions, Acronyms, Abbreviations

FP: Function Points
ILF: Internal logic file
EIF: External interface file
EI: External Input
EO: External Output
EQ: External Inquiries
DD: Design document
Visitor: a generic person who access the website or the app without being authenticated
User: equivalent to registered user, a visitor who already performed the registration and has now access to the system
Car: with this term we always refer to one of our cars
Car tag: a term to label the status of a car, it can be "available" or "reserved" or "in use"

Available car: a car that is currently not in use nor reserved by anyone and has at least 30% of battery level
Reservation: it can be requested by a user for an available car, this gives the user the possibility to unlock the reserved car within a time span of 1 hour from the reservation time, after this time the reservation expires
Reserved car: a car that can be unlocked only by the one who reserved it
Ignition code: this code is sent to the user when a reservation is successful, user will have to type this code on the OBS lock screen in order to ignite the engine
Locking a car: after the passengers are all out of the car and the door are all closed the system counts 40 seconds and then locks the car. There are two types of locking: if the user wants to keep the car on pause, the car is locked and it remains on pause for that user, otherwise the car is locked and made available to other users. System locks the car only if the car is inside the Safe Area
Pause: when the user turns off the engine the systems asks him on the OBS if he wants to put the car on pause, if the user taps "YES" the pause state is applied: the user keeps paying the per-minute fee until he unlocks it again
Ride or Trip: it starts when the engine is ignited or 2 minutes after the car is unlocked, and ends when the car is locked
Temporary bill: keeps track of the duration of the trip or the pause to calculate the amount of money to pay
Safe area: we consider a safe area the set of every possible legal parking spot within The Area, so all the places where it is possible to park the car without being fined
SPA: these are Special Parking Areas where cars can be parked and also recharged through a special plug
The System: the personification of the software, which manages all the functionalities of PowerEnJoy
MSM: Money Saving Mode, this can be enabled in the car after selecting a destination, this ensures to find parking solution to get a discount, optimizing the cost of the trip; the solution takes also into account the position of cars in The Area to ensure a uniform distribution
OBS: On Board Screen, it's the display inside the car, where the user can visualize informations about the car and the trip

1.4 Reference Documents

- The project documentation: RASD & DD & ITPD
- Project Management Basics + Advanced lecture material
- Project planning example document.pdf
- Project assignment
- http://www.qsm.com/resources/function-point-languages-table
- http://www.softwaremetrics.com/pdf
- COCOMO II - Model Definition Manual, USC, version 2.1

2 Project size, cost and effort estimation

2.1 Size estimation: function points

The Function Points Analysis will be our tool to estimate the complexity of each functionality in the PowerEnJoy system and the size of the Data we have to deal with. The complexity of the user interface will not be discussed in this document because the focus here is on the business logic and not on the presentation layer.

We will assign to each function and data set a complexity level (Low or Average or High) depending on the amount of data used and on the functionality provided, and then, each component type will have a Weight multiplier corresponding to each level of complexity.

The reference table for all the weight multiplier is reported below, and for simplicity the weight multipliers are also reported in the sub-chapters.

Unadjusted Function Points Table

	Complexity Weight		
Function Type	*Low*	*Avg*	*High*
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

2.1.1 Internal Logical Files (ILFs)

This chapter is dedicated to the data set internally managed by the system. All the data are stored in a relational database, so they will be presented by use of a Logical Schema, where all the attributes of each table are shown. This is needed in order to find the complexity of each entity in the database, according to the following table.

Complexity (and Weight) Reference Table for ILF

Record Elements	**Data Elements**		
	1-6	*7-12*	*≥13*
1	Low (7)	Low (7)	Avg (10)
2-3	Low (7)	Avg (10)	High (15)
≥4	Avg (10)	High (15)	High (15)

Here follows the Logical Schema of the entities in our database:

User (DrivingLicence, Name, Surname, Email, Password, BirthDate, PaymentData)
Staff (ID, Name, Surname, Email, Password)
HelpRequest (ID, UserID, StaffID, Date, CarID*, Description*)
Reservation (ID, UserID, CarID, Date, Time)
Car (ID, State, Position, BatteryLevel)
Bill (ID, UserID, Amount, Date, CarID, TimeStart, TimeEnd)
SPA (ID, TotalChargingTower, AvailableChargingTower)

Entity Complexity

ILF	**Records**	**Datas**	**Complexity**	**FPs**
UserTable	1	7	Avg	10
StaffTable	1	5	Low	7
HelpRequest	1	6	Low	7
Reservation	1	5	Low	7
Car	1	4	Low	7
Bill	1	7	Avg	10
SPA	1	3	Low	7
			Total:	**55**

2.1.2 External Interface Files (EIFs)

In this chapter we have to estimate the complexity of the part of our system that manages external data. In our case the data provided by external applications consist in the maps from Here Maps, the data of the external map about safe areas and SPA and the data retrieved from CarSensors.

Since this external components offer simple interfaces to access all the functionalities, the interaction with them doesn't have a big impact on the complexity of our system, except for what concerns the HERE Maps, because it's easy to use the interface but the integration of the component in the application and in the car will require some more work in terms of coding.

Interface Complexity

EIF	Complexity	FPs
HERE Maps	High	10
External Map	Low	5
CarSensors	Low	5
	Total:	**20**

2.1.3 External Inputs (EIs)

We will analyze the complexity of all the functions dealing with data coming from the external environment. This basically includes all the operations that can be performed by the clients through their application. A complexity will be associated to each operation, with a reasonable explanation, and the weight in Function Points can be found in the following reference table.

Complexity	FPs
Low	3
Avg	4
High	6

PowerEnJoy application provides several functions interacting with the system, this items in the list are obtained by looking at the goals of our project and we will present them separated by the actor who can perform them.

User:

- **Login/Logout**: these operations are quite simple and involve only the AccessManager, they can both be accomplished with low complexity code
- **Register new account:** the register operation has an average complexity because it must check all data fields and there are many conditions to cover
- **Car reservation:** average complexity. It is composed by a sequence of sub-operations that have to be well integrated
- **Retrieving car information**: high complexity. This includes all the kind of data that can be retrieved by the user from the web or mobile application
- **Car unlock:** high complexity. Many software and hardware components are involved and distance check is necessary
- **Pause a trip**: high complexity, the choice of the user has to be evaluated and this action will be followed by a payment and the future status of the car will depend on this operation
- **Make a help request:** average complexity. The function is executed by the HelpRequestManager and it requires to add a request in a queue

Staff:

- **Manage help requests**: high complexity. It uses HelpRequestManager to pop a request from a queue and it also requires the instantiation of a connection with the user

EIs	**FPs**
Login	3
Logout	3
Register new account	4
Car reservation	4
Retrieve car information	6
Car unlock	6
Pause a trip	6
Make a helpRequest	4
Manage help requests	6
Total:	**42**

2.1.4 External Outputs (EOs)

These operations are internal to our system and produce data for the external environment. The complexity multiplier for this kind of components are reported in the following table:

Complexity	FPs
Low	4
Average	5
High	7

The operations associated to this category are the following:

- **Invoices:** high complexity**.** The application allows the creation of invoices. This is a quite complex operation as it needs to collect informations from IFLs and EFLs.
- **Visualization of remaining battery on the car:** low complexity**.** The System let the user know continuously the remaining battery charge of his car. This is a simple activity since the information is taken from the car and forwarded to the user.

EOs	FPs
Invoices	7
Battery on the car	4
Total:	**11**

2.1.5 External Inquiries (EQs)

These are elementary processes that involve both input and output components and consist in data retrieval from one or more ILFs and EIFs. It must be noted that in this processes no ILF is updated and no derived data is contained in the output.

Complexity	FPs
Low	3
Average	4
High	6

- **Search cars:** this is a high complexity operation because many components are involved.
- **Search SPA:** high complexity. It requires the algorithm to run a search based on the data in the database

- **MSM**: high. This is our proprietary algorithm to try to minimize the cost of a trip while keeping a quite uniform distribution of the car in the city
- **User views his bill history:** low complexity, it is a straightforward operation.
- **Staff views reservation's history:** low complexity, it is a straightforward operation.
- **Staff views car status on map**: high. All the cars have to be retrieved from the database, with their information attached and their position has to be marked on a map

EQs	FPs
Search cars	6
Search SPA	6
MSM	6
View bill history	3
View reservation History	3
View car status	6
Total:	**30**

2.1.6 Overall estimation

Function Type	FPs
ILF	55
EIF	20
EI	42
EO	11
EQ	30
Total:	**158**

QSM Function Points Languages Table

Language	QSM SLOC/FP			
	Low	***Medium***	*High*	*Avg*
J2EE	14	**49**	67	46

Referring to the tables above we can have an estimate of the logical lines count in our project. We chose to use the median as gearing factor because it is a more accurate indicator of the central tendency, and our results are:

$$\mathbf{SLOC_M} = 158 \times 49 = 7742$$

with an upper bound of:

$$\mathbf{SLOC_H} = 158 \times 67 = 10586$$

2.2 Cost and effort estimation: COCOMO II

2.2.1 Scale Drivers

In order to better evaluate the scale drivers of our project we will use the Official COCOMO II table for Scale Factor Values:

Scale Factors (*Wi*)	**Very Low**	**Low**	**Nominal**	**High**	**Very High**	**Extra High**
PREC	thoroughly unprecedented **6.20**	largely unprecedented 4.96	somewhat unprecedented 3.72	generally familiar 2.48	largely familiar 1.24	thoroughly familiar 0.00
FLEX	rigorous 5.07	occasional relaxation 4.05	some relaxation **3.04**	general conformity 2.03	some conformity 1.01	general goals 0.00
RESL[a]	little (20%) 7.07	some (40%) 5.65	often (60%) 4.24	generally (75%) **2.83**	mostly (90%) 1.41	full (100%) 0.00
TEAM	very difficult interactions 5.48	some difficult interactions 4.38	basically cooperative interactions 3.29	largely cooperative 2.19	highly cooperative **1.10**	seamless interactions 0.00
PMAT	Weighted average of "Yes" answers to CMM Maturity Questionnaire					
	SW-CMM Level 1 lower 7.80	SW-CMM Level 1 upper 6.24	SW-CMM Level 2 **4.68**	SW-CMM Level 3 3.12	SW-CMM Level 4 1.56	SW-CMM Level 5 0.00

The **Precedentedness (PREC)** for a project reflects if the team has yet developed a similar project. For our project this driver is set to Very Low because our team never developed a system of this dimension and in this area.

The **Flexibility (FLEX)** of a system measures if there are strict constraints to conform in the requirements or external interfaces specs. In The System the flexibility is set to Nominal because there are some external interfaces to deal with and the requirements were sufficiently strict but in the end we are rather free to decide how to implement TS and which technology to use.

The **Architecture/Risk resolution (RESL)** is made to control if we have a good risk management plan, definition of budget and schedule and focus on architectural definition. In our case we set this parameter to High because our risk management plan is strong and our schedule is thorough.

The **Team cohesion (TEAM)** measures the ability of the members to work and collaborate in a team and to share the same vision on the project. This driver is Very high in our case.

The **Process maturity (PMAT)** refers to the CMMI: a method for assessing the maturity of a software organization. This project is evaluated at a CMM level 2, so with a Nominal value for the driver, because this is our first project but we achieved all the goals.

The results of the scale drivers evaluation is the following:

Scale Driver:	**Factor**	**Value**
PREC	Very Low	6.20
FLEX	Nominal	3.04
RESL	High	2.83
TEAM	Very high	1.10
PMAT	Nominal	4.68
	Total (SF):	**17.85**

2.2.2 Cost Drivers

2.2.2.1 Product factors

2.2.2.1.1 Required software reliability(RELY)

A malfunctioning in our software system will lead, in the worst case, to a car theft, and since PowerEnJoy has invested much money to buy the cars, this is considered to be a large financial loss. For this reason this factor is set to High.

RELY descriptors	slight inconvenience	low, easily recoverable losses	moderate, easily recoverable losses	high financial loss	risk to human life
Rating levels	Very Low	Low	Nominal	High	Very High
Effort multipliers	0.82	0.92	1.00	**1.10**	1.26

2.2.2.1.2 Database size (DATA)

This parameter measures the ratio of the dimension of the testing database (in bytes) to the SLOC in the entire program. We don't know yet the database dimensions but we can estimate a 2 MB testing database and an approximate 8000 SLOC for the project. Using these data we have a D/P factor of 250; the result of this driver level being High.

DATA Descriptors	Testing DB bytes/Pgm SLOC<10	10≤D/P< 100	100≤D/P< 1000	D/P≥1000
Rating Levels	Low	Nominal	High	Very High
Effort Multipliers	0.90	1.00	**1.14**	1.28

2.2.2.1.3 Product complexity (CPLX)

Complexity is divided into five areas: control operations, device-dependent operations, data management operations and user interface management operations. For the control operations we have a nominal complexity, computational operations is low complexity, for device-dependent operations we have high complexity, data management operations is nominal complexity and for user interface management operations we have a nominal complexity. In the end we have a Nominal complexity for this driver.

Rating Level	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.73	0.87	**1.00**	1.17	1.34	1.74

2.2.2.1.4 Required reusability (RUSE)

With our approach of giving a limited responsibility to the components we are going to develop components with no code duplication, but that can have a limited reusability, especially across other programs and further, on the other side they can work together with more than one component within our system, so we will rate this as Nominal.

RUSE Descriptors	None	Across Project	Across Program	Across Product Line	Across Multiple Product Lines
Rating Levels	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.95	**1.00**	1.07	1.15	1.24

2.2.2.1.5 Documentation match to life-cycle needs (DOCU)

This will rate how the project documentation covers it's life cycle need, and in our case we produced a documentation covering the whole life-cycle, but not extra phases, so a Nominal level is adequate.

DOCU Descriptors (life-cycle needs)	Many uncovered	Some uncovered	Right-sized	Excessive	Very excessive
Rating Levels	Very Low	Low	Nominal	High	Very High
Effort Multipliers	1.29	1.12	**1.00**	0.90	0.81

2.2.2.2 Platform factors

2.2.2.2.1 Execution time constraint (TIME)

The TIME rating is based on the percentage of available execution time expected to be used: our system will run and be reachable 24/24 hours, but since it manages cars we expect to have peak of usage only in some hours of the day, and a moderate usage for the rest of the day, and as an average we can state that the rating is High.

TIME Descriptors (% use of available execution time)	50%	70%	85%	95%
Rating Levels	Nominal	High	Very High	Extra High
Effort Multipliers	1.00	**1.11**	1.29	1.63

2.2.2.2.2 Storage constraint (STOR)

This rating reflects the percentage of main storage expected to be used by the software product and any subsystems, and since our storage solution is cloud, we have virtually unlimited storage available, so the value for this will be Nominal.

STOR Descriptors (% use of available storage)	50%	70%	85%	95%
Rating Levels	Nominal	High	Very High	Extra High
Effort Multipliers	**1.00**	1.05	1.17	1.46

2.2.2.2.3 Platform volatility (PVOL)

The complex of hardware and software is expected to go through very few changes during its life, with an exception for eventual bug and vulnerabilities fixes, so we can rate the volatility as Low.

PVOL Descriptors	Major change every 12 months, minor change every month	Major change every 6 months, minor change every 2 weeks	Major change every 2 months, minor change every week	Major change every 2 weeks, minor change every 2 days
Rating Levels	Low	Nominal	High	Very High
Effort Multipliers	**0.87**	1.00	1.15	1.30

2.2.2.3 Personnel Factors

2.2.2.3.1 Analyst capability (ACAP)

For what concerns the analysis, we rate our capabilities as Nominal, because the requirement analysis will be done taking some domain assumptions and the design will not go too much into details.

ACAP Descriptors (Percentile)	15th	35th	55th	75th	90th
Rating Levels	Very Low	Low	Nominal	High	Very High
Effort Multipliers	1.29	1.12	**1.00**	0.90	0.81

2.2.2.3.2 Programmer capability (PCAP)

We are quite familiar with good coding practices, and we know how a team has to communicate and cooperate to produce quality software, so we can rate our capabilities as High.

PCAP Descriptors (Percentile)	15th	35th	55th	75th	90th
Rating Levels	Very Low	Low	Nominal	High	Very High
Effort Multipliers	1.29	1.12	1.00	**0.90**	0.81

2.2.2.3.3 Personnel continuity (PCON)

There should be no reason to change the team members during the development of this project, so our annual turnover is the smallest possible and the continuity is Very High.

PCON Descriptors (Turnover per year)	48%	24%	12%	6%	3%
Rating Levels	Very Low	Low	Nominal	High	Very High
Effort Multipliers	1.29	1.12	1.00	0.90	**0.81**

2.2.2.3.4 Application experience (APEX)

In the past we developed a desktop application with a client-server architecture, but in this project we are facing a more complex system, so we value our experience as Low.

APEX Descriptors	≤2 months	6 months	1 year	3 years	6 years
Rating Levels	Very Low	Low	Nominal	High	Very High
Effort Multipliers	1.19	**1.09**	1.00	0.91	0.85

2.2.2.3.5 Platform experience (PLEX)

The JEE platform is totally new to us but we already have a basic knowledge on how a complex system with server and database could work, this results in a Low rating.

PLEX Descriptors	≤2 months	6 months	1 year	3 years	6 years
Rating Levels	Very Low	Low	Nominal	High	Very High
Effort Multipliers	1.19	**1.09**	1.00	0.91	0.85

2.2.2.3.6 Language and tool experience (LTEX)

We have some experience with Java and with Eclipse, but our previous project were smaller and didn't involve hardware components, so we will rate our experience as Low.

LTEX Descriptors	≤2 months	6 months	1 year	3 years	6 years
Rating Levels	Very Low	Low	Nominal	High	Very High
Effort Multipliers	1.20	**1.09**	1.00	0.91	0.84

2.2.2.4 Project Factors

2.2.2.4.1 Use of software tools (TOOL)

The software we are going to design will reach a strong maturity level, and will go through a full life-cycle, so we can rate the TOOL as High.

TOOL Descriptors	Edit, code, debug	Simple, frontend, backend CASE, little integration	Basic life-cycle tools, moderately integrated	Strong, mature life-cycle tools, moderately integrated	Strong, mature, proactive life-cycle tools, well integrated with processes, methods, reuse
Rating Levels	Very Low	Low	Nominal	High	Very High
Effort Multipliers	1.17	1.09	1.00	**0.90**	0.78

2.2.2.4.2 Multisite development (SITE)

For this rating we will do the average of site collocation and communication support. We consider a same metro area collocation because we live in different cities but not too far from each other. The development process was mixed, with some physical meeting in the same place and some VoIP calls, we also worked on a cloud platform so wideband was fundamental for us, the average with this descriptors is High.

SITE: Collocation Descriptors	International	Multi-city and Multi-company	Multi-city or Multi-company	Same city or metro area	Same building complex	Fully collocated
SITE: Communications Descriptors	Some phone, mail	Individual phone, FAX	Narrow band email	Wideband electronic communication	Wideband elect. comm., occasional video conf.	Interactive multimedia
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.22	1.09	1.00	**0.93**	0.86	0.80

2.2.2.5 General Factor

2.2.2.5.1 Required development schedule (SCED)

Due to the small dimension of our team we opted for a stretch-out instead of an acceleration in the schedule, with respect to a nominal schedule. A stretch of 130% is rated as a High schedule constraint.

SCED Descriptors (w.r.t nominal)	75%	85%	100%	130%	160%
Rating Level	Very Low	Low	Nominal	High	Very High
Effort Multiplier	1.43	1.14	1.00	**1.00**	1.00

2.2.3 Effort equation

Cost Driver	**Value**
RELY	1.10
DATA	1.14
CPLX	1.00
RUSE	1.00
DOCU	1.00
TIME	1.11
STOR	1.00
PVOL	0.87
ACAP	1.00
PCAP	0.90
PCON	0.81
APEX	1.09
PLEX	1.09
LTEX	1.09
TOOL	0.90
SITE	0.93
SCED	1.00
Product (EAF):	**0.956912545**

Here we use the Estimation Equations from the COCOMO II Manual

EAF = 0.956912545

A = 2.94

B = 0.91

SF = 17.85

E = B + 0.01 x SF = 1.0885

$kSLOC_M$ = 7.742

EFFORT = A x EAF x $(kSLOC_M)^E$ = **26.11 PM**

If we consider the upper bound $kSLOC_H$ = 10.586, we get 36.70 PM

2.2.4 Schedule estimation

An estimate of the schedule can be found with this additional equations from the Manual:

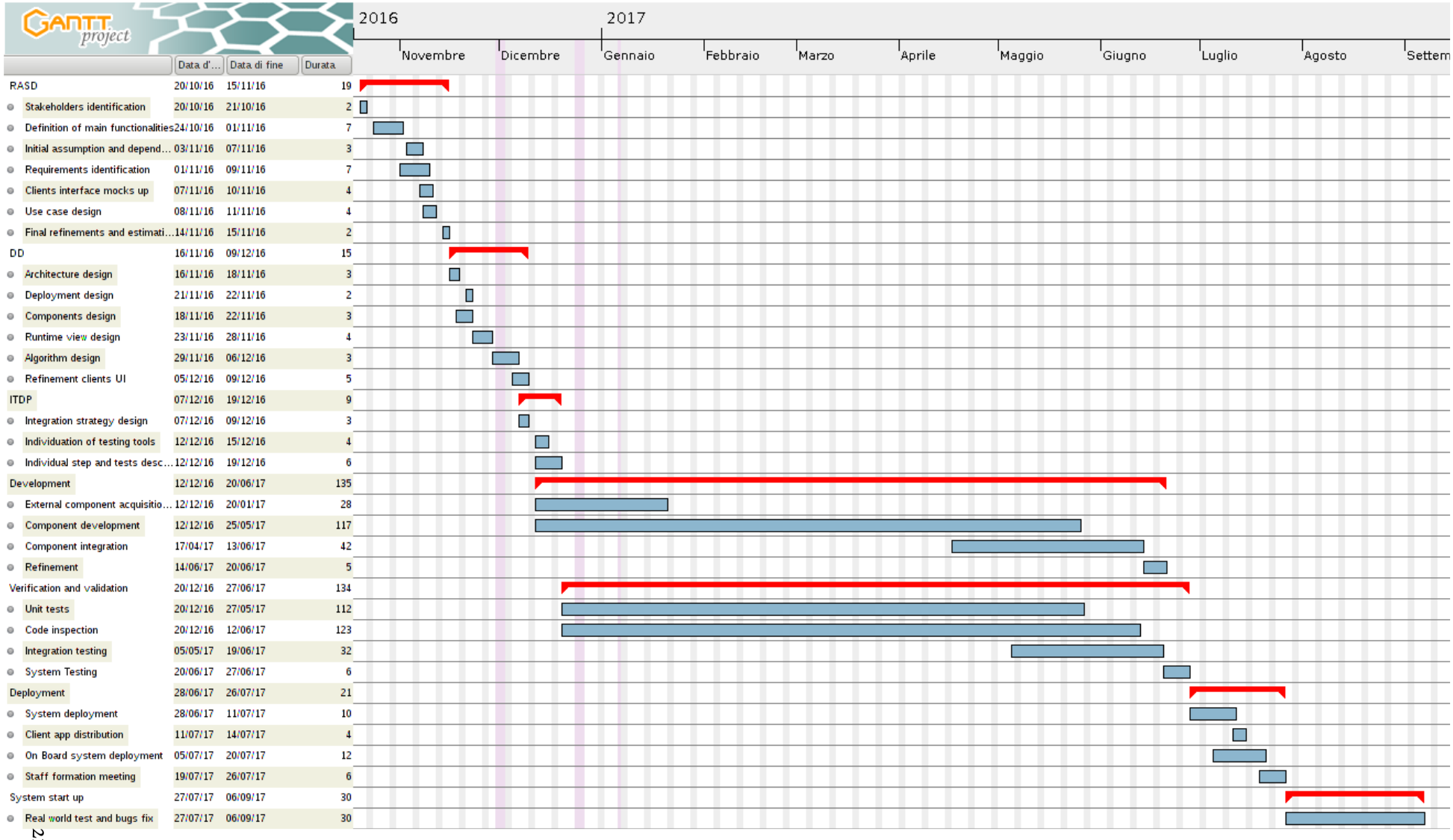
F = 0.28 + 0.2 x (E - B) = 0.3157

DURATION = 3.67 x $EFFORT^F$ = **10.28 months**

If we consider the upper bound 36.70 PM, we get 11.45 months

3 Schedule

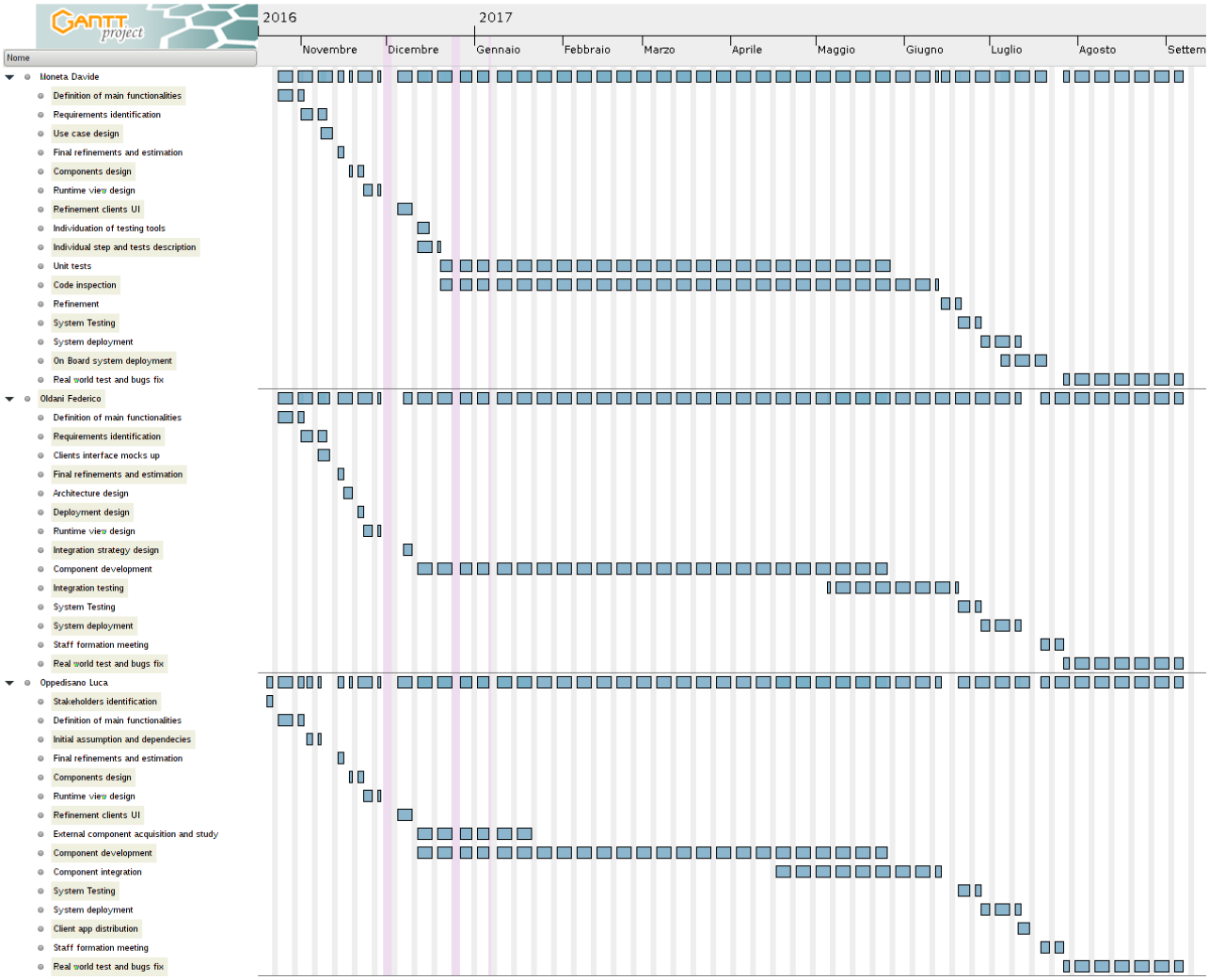
In this paragraph we will present a high level project schedule. The whole project takes about ten months (design and development)


GANTT project
2016
2017
Novembre
Dicembre
Gennaio
Febbraio
Marzo
Aprile
Maggio
Giugno
Luglio
Agosto
Settem
Data d'...
Data di fine
Durata
RASD 20/10/16 15/11/16 19
Stakeholders identification 20/10/16 21/10/16 2
Definition of main functionalities 24/10/16 01/11/16 7
Initial assumption and depend... 03/11/16 07/11/16 3
Requirements identification 01/11/16 09/11/16 7
Clients interface mocks up 07/11/16 10/11/16 4
Use case design 08/11/16 11/11/16 4
Final refinements and estimati... 14/11/16 15/11/16 2
DD 16/11/16 09/12/16 15
Architecture design 16/11/16 18/11/16 3
Deployment design 21/11/16 22/11/16 2
Components design 18/11/16 22/11/16 3
Runtime view design 23/11/16 28/11/16 4
Algorithm design 29/11/16 06/12/16 3
Refinement clients UI 05/12/16 09/12/16 5
ITDP 07/12/16 19/12/16 9
Integration strategy design 07/12/16 09/12/16 3
Individuation of testing tools 12/12/16 15/12/16 4
Individual step and tests desc... 12/12/16 19/12/16 6
Development 12/12/16 20/06/17 135
External component acquisitio... 12/12/16 20/01/17 28
Component development 12/12/16 25/05/17 117
Component integration 17/04/17 13/06/17 42
Refinement 14/06/17 20/06/17 5
Verification and validation 20/12/16 27/06/17 134
Unit tests 20/12/16 27/05/17 112
Code inspection 20/12/16 12/06/17 123
Integration testing 05/05/17 19/06/17 32
System Testing 20/06/17 27/06/17 6
Deployment 28/06/17 26/07/17 21
System deployment 28/06/17 11/07/17 10
Client app distribution 11/07/17 14/07/17 4
On Board system deployment 05/07/17 20/07/17 12
Staff formation meeting 19/07/17 26/07/17 6
System start up 27/07/17 06/09/17 30
Real world test and bugs fix 27/07/17 06/09/17 30

4 Resource allocation

Task	Effort (person-days)	Duration (days)	Dependencies
RASD			
1.Stakeholders identification	4	2	
2.Definition of main functionalities (include stakeholders meeting)	10	7	A1
3.Initial assumption and dependencies	3	3	A2
4.Requirements identification	10	7	A2
5.Clients interface mocks up	4	4	A2
6.Use case design	6	4	A4
7.Final refinements	4	2	All activities before
DD			RASD
8.Architecture design	3	3	
9.Deployment design	2	2	A8
10.Components design	5	3	
11.Runtime view design	10	4	A8, A10
12.Algorithm design	5	3	
13.Refinement clients UI	8	5	
ITPD			
14.Integration strategy design	3	3	A10
15.Individuation of testing tools	3	4	A14
16.Individual step and tests	10	6	A14

description			
Development			**DD**
17.External component acquisition and study	18	28	
18.Component development	80	117	
19.Component integration	35	42	A17
20.Refinement	10	5	A17, A18, A19
Verification and Validation			**ITPD**
21.Unit tests	75	112	Together with A18
22.Code inspection	80	123	Together with A18, A19
23.Integration testing	29	32	Together with A19
24.System Testing	15	6	All activities before
Deployment			All activities before
25.System deployment	15	10	
26.Client app distribution	4	4	A25
27.On Board system deployment	12	12	
28.Staff formation meeting	6	6	
System start up			All activities before
29.Real world test	55	30	



Notice that in case of overlaps in resource activity, it means that the resource will spend 50% of his working hours in one activity and 50% of his working hours in the other activity.

5 Risk management

As part of the planning process we have to take into account any possible situation that may occur during the realization of the project, so we will identify all the risks we may face, giving also directives on how to minimize the possibility to incur in this risks, and planning a strategy to mitigate the negative effect of those situation if they cannot be avoided.

Risk	Prevention strategy	Mitigation strategy
Malfunctioning of hardware components needed for the implementation	Use new hardware and treat it well	New hardware has to be bought or rented as soon as possible
Doubts on some decisions to take	Ask to have specifications that leave no doubts	Ask to the stakeholders or to other people who deal with this kind of projects or take a pool between the project members
Financial and political stakeholders may change idea about some legal or technical matter	Make unchangeable agreements on all those matters considered core requirements	Meet with the stakeholders as soon as possible to find a compromise
Legislation may change at national or higher level	Try to avoid situation that are currently in a political debate	Meet with the stakeholders and find the solution that adapts to the new legislation with less impact on the existing project
Users may reject the new service for different reasons (they don't need it or they prefer the competitors)	Invest in advertisement campaigns, offer launch promotions, predict user needs	Ask the people what they think may improve the service
Personnel shortfall, members may not have enough workforce, members may quit without warning	Be careful when doing estimations on the effort and cost of the project and allocate resources so that each part of the project is familiar to more than one member	Talk to the stakeholder to reduce the load of the project
Capability shortfall, members knowledge may be overestimated	Pre evaluate the knowledge of each team member to adapt the complexity of the project to the actual capabilities	Talk to the stakeholders to find a way to simplify the project, maybe removing some features
Permanent loss of project material	Use cloud storage, loss risk is almost zero	Talk to the stakeholders to extend the deadlines because part of the project has to be

		rewritten
Any change in external components' terms and conditions or API	Choose external components considered reliable, look at the terms, implement a possibly decoupled interaction	Choose between switching to another components or adapting the implementation to the changes
Any change in mobile phone company data plan used for the on board system	Try to find an agreement with the phone company that will not change	Choose between the migration to another company or accept the changes
Unrealistic schedule	Use previous experience to plan a reasonable schedule, insert some slack between tasks to provide extra time	Talk with the stakeholders to extend the deadline or to negotiate the requirements and simplify the project
Unrealistic budget	Do an accurate cost estimation before proposing something	Talk to the stakeholders to increase the budget or to find something to cut to reduce the cost
Wrong functionality or user interface	Make sure that the goals of the project are clear to both the stakeholders and the project members, avoid making things different than agreed just for personal taste	If this functionality has emerged it may fit the system better than the correct one, see if the stakeholders can accept it, otherwise it has to be re implemented
Goldplating	Team members should avoid investing time in something that doesn't falls to them, the granularity deepness to reach must be clear at the beginning	Consult other team members and the stakeholders to decide if the extra work has to be included or not in the project
Requirements volatility	Try to have a full picture of the project at the beginning so eventual problems can be found earlier and requirements can be changed with low impact, develop decoupled components	Talk to the stakeholders to re arrange the deadlines for the new requirements

6 Hours of work

6.1 Davide Moneta

15/01/2017	1h30’
16/01/2017	1h00’
18/01/2017	2h30’
19/01/2017	3h45’
21/01/2017	3h00’

6.2 Federico Oldani

17/01/2017	1h30’
18/01/2017	1h00’
19/01/2017	3h00’
20/01/2017	4h00’
21/01/2017	3h00’
22/01/2017	1h00’

6.3 Luca Oppedisano

15/01/2017	4h00’
18/01/2017	4h30’
19/01/2017	4h30’
20/01/2017	5h30’
21/01/2017	5h00’